

Esame di Calcolatori Elettronici T

13 Gennaio 2016 (Ing. Informatica)

Esercizio 1

Progettare un sistema basato sul microprocessore DLX (**clock a 1 GHz**) dotato di **320 MB di EPROM** mappata negli indirizzi bassi e **128 MB di RAM** mappata negli indirizzi alti. Nel sistema sono presenti **due periferiche a 8 bit**, già progettate, denominate **INPUT_PORT**, in grado di ricevere dati dall'esterno utilizzando il protocollo di *handshake*, e **OUTPUT_PORT**, in grado di inviare dati all'esterno utilizzando il protocollo di *handshake*. **Ogni 10 ms**, una rete da progettare denominata **TIMER**, deve generare una richiesta di interrupt. Durante la gestione di tale periodica richiesta di interruzione, mediante opportuni comandi software, deve poter essere determinabile se è possibile leggere un carattere da **INPUT_PORT**. In caso affermativo, deve essere eseguita una lettura da **INPUT_PORT** e, nel caso sia anche possibile scrivere in **OUTPUT_PORT**, inviare **contemporaneamente** il dato letto da **INPUT_PORT** al dispositivo **OUTPUT_PORT**. Al contrario, nel caso non sia possibile leggere da **INPUT_PORT**, l'interrupt handler non deve eseguire alcun trasferimento con le periferiche **INPUT_PORT** e **OUTPUT_PORT**.

- Evidenziare e risolvere eventuali criticità
- Indicare i segnali di chip select di tutte le periferiche, memorie e di eventuali altri segnali presenti nel sistema progettato
- Progettare, **ottimizzando le risorse utilizzate**, tutte le reti necessarie al funzionamento del sistema (inclusa la rete **TIMER**) indicando chiaramente le connessioni di tutte le periferiche e le memorie ai bus del DLX
- Scrivere il codice dell'*interrupt handler* che gestisce l'interruzione generata dal **TIMER** in accordo alle specifiche del problema. Si assuma che i registri da **R20** a **R30** possano essere utilizzati senza la necessità di dover preservare il loro contenuto

Esercizio 2

Indicare quali strategie, hardware e software, è possibile adottare per risolvere eventuali alee generate dalla sequenza di istruzioni seguente:

ADD R7, R8, R9

ADD R1, R2, R3

SUB R4, R5, R6

SUB R10, R7, R20

ADDI R25, R20, 8

FORWARDING UNIT?
CRITICO

Esercizio 3

Descrivere il principio di funzionamento del Branch Target Buffer

MAPPING

EPROM 320 KB : EPROM 256 KB : $0 \times 00000000 \rightarrow 0 \times 0FFFFFFF$ (4×64)

EPROM 64 KB : $0 \times 10000000 \rightarrow 0 \times 13FFFFFF$ (4×16)

RAM 128 KB : $0 \times F8000000 \rightarrow 0 \times FFFFFFFF$ (4×32)

CS_EPROM_L_0 = BA31 BA28 BE 0

CS_EPROM_L_1 = BA31 BA28 BE 1

CS_EPROM_L_2 = BA31 BA28 BE 2

CS_EPROM_L_3 = BA31 BA28 BE 3

CS_EPROM_H_0 = BA31 BA28 BE 0

CS_EPROM_H_1 = BA31 BA28 BE 1

CS_EPROM_H_2 = BA31 BA28 BE 2

CS_EPROM_H_3 = BA31 BA28 BE 3

CS_RAM_0 = BA31 BA30 BE 0

CS_RAM_1 = BA31 BA30 BE 1

CS_RAM_2 = BA31 BA30 BE 2

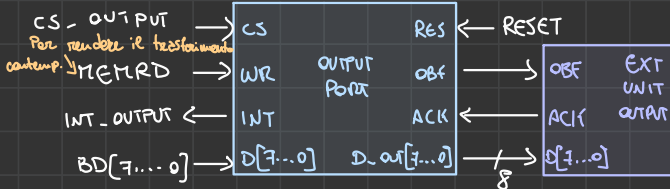
CS_RAM_3 = BA31 BA30 BE 3

CS_INPUT = BA31 BA30 BE 0

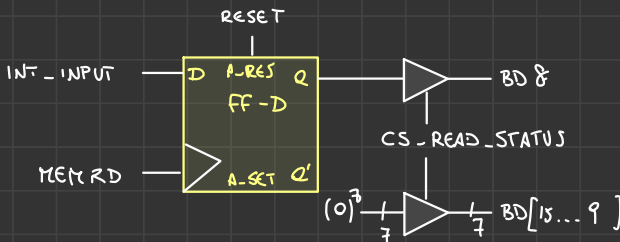
CS_OUTPUT = CS_INPUT · OUTPUT_READY

CS_READ_STATUS = BA31 BA30 BE 1

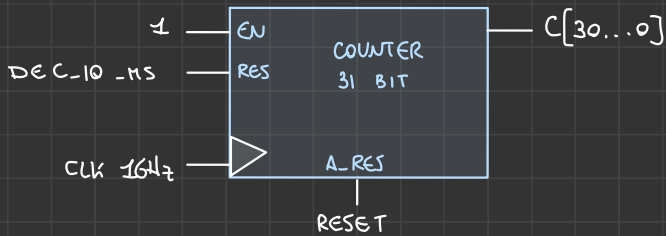
PORTE I/O



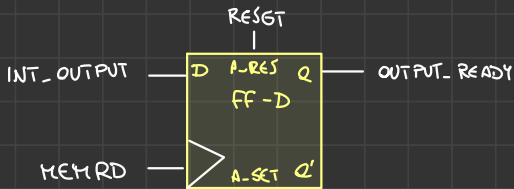
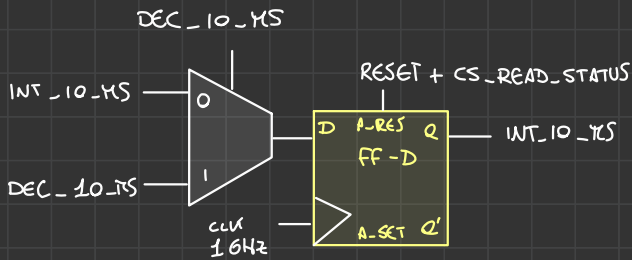
INT (TO DLX) = 1 μS



TIMER



$$\text{DEC_10_MS} = 0 \times 0098967F$$



CODING

```

0h  LHI  R20, 0x8000
4h  LBU  R21, 0x0001 (R20)
8h  BNEZ  R21, INPUT
Ch  RFE;
10h  INPUT  LBU  R25, 0x0000 (R20)
14h          RFE;
18h
1ch

```

INPUT = 4h